published in J. of Computation and Applied Mathematics (JCAM), Vol. 156, 403-432, 2003.

TEN METHODS TO BOUND MULTIPLE ROOTS OF POLYNOMIALS

SIEGFRIED M. RUMP*

Abstract. Given a univariate polynomial P with a k-fold multiple root or a k-fold root cluster near some $\widetilde{z}$, we discuss various different methods to compute a disc near $\widetilde{z}$ which either contains exactly or contains at least k roots of P. Many of the presented methods are known, some are new. We are especially interested in rigorous methods, that is taking into account all possible effects of rounding errors. In other words every computed bound for a root cluster shall be mathematically correct. We display extensive test sets comparing the methods under different circumstances. Based on the results we present a hybrid method combining five of the previous methods which, for given $\widetilde{z}$, i) detects the number k of roots near $\widetilde{z}$ and ii) computes an including disc with in most cases a radius of the order of the numerical sensitivity of the root cluster. Therefore, the resulting discs are numerically nearly optimal.

1. Introduction and notation. Throughout the paper denote by $P = \sum_{\nu=0}^{n} p_\nu z^\nu \in \mathbf{C}[z]$ a (real or complex) polynomial. Let $\widetilde{z} \in \mathbf{C}$ be given such that P has some k roots near $\widetilde{z}$. The k roots may be clustered or multiple, or just one k-fold root. We do not require a priori assumptions on the multiplicity, distribution or distance of the roots of P from $\widetilde{z}$; so $\widetilde{z}$ and k are rather to be understood as a guess.

The problem is to find a disc $D(c,r) := \{z \in \mathbf{C} : \ |z-c| \le r\}$ containing either exactly k roots of P or at least k roots of P. The midpoint c is usually near $\widetilde{z}$, and the radius r should be as small as possible. There is a huge literature devoted to polynomial root finding; for an overview see, for example, [21]. However, many publications are concerned with simple roots or even require all roots to be pairwise distinct. The purpose of the present paper is two-fold. First, to collect some representative results for multiple roots, to present some new methods for this problem and finally to describe a hybrid method delivering almost optimal results. Second, the computed numerical results shall be correct, correct in a mathematical sense including estimation of all possible rounding and computational errors. The methods shall exclusively use ordinary floating point arithmetic and no higher precision to achieve a performance not too far from a numerically approximating algorithm.

In our experience approximations of the roots calculated by some standard numerical algorithm are very accurate, with errors of the order of the numerical sensitivity (see Section 3). To that extent one may ask why rigorous bounds for a root cluster are interesting or necessary at all. For the "usual" practical application we think they are not. However, beside being an interesting *mathematical* question to compute bounds on a digital computer which are *mathematically* correct, there is recent interest in so-called computer assisted proofs. Famous examples are the proof of the almost 400-years-old Kepler conjecture [13], the double bubble conjecture [14], the proof of existence of Lorenz-attractors [35], and more. For an overview cf. [11]. The common principle of those methods is that the problem is transformed into a nonlinear optimization problem, a root-finding problem of certain systems of nonlinear equations and alike. Needless to say that for the rigor of a mathematical proof those numerical problems are to be solved rigorously. This is done by so-called self-validating methods [31], and by all methods presented in this paper.

The appealing of those methods is that they exclusively use floating point arithmetic and no (simulated) higher precision. Thus they are very fast, only by a moderate factor slower than traditional numerical algorithms (without verification). Given approximations of the roots of a polynomial computed by some standard numerical algorithm, all our methods compute rigorous bounds for a cluster of roots in some $\mathcal{O}(n^2)$ operations.

* Technical University Hamburg-Harburg, Schwarzenbergstr. 95, 21071 Hamburg, Germany

For given $c \in \mathbf{C}$ we may expand P at c resulting in

$$Q = P(c+z) = \sum_{\nu=0}^{n} q_\nu z^\nu = \sum_{\nu=0}^{n} \frac{P^{(\nu)}(c)}{\nu!} z^\nu. \tag{1}$$

In case c is a k-fold root of P, $q_0 = \cdots = q_{k-1} = 0$. Hence we may expect q_ν to be small in absolute value for $0 \le \nu \le k-1$. A common Ansatz considers

$$\overline{Q}(z) := \sum_{\nu=k}^{n} q_\nu z^\nu \tag{2}$$

with an exact k-fold root at the origin and estimates the change of roots on a homotopy path from $\overline{Q}$ to Q. As has been mentioned we will investigate the behavior of the different methods under rigorous control of rounding errors. For example, for given P with coefficients being floating point numbers, the coefficients of Q as in (1) are generally not exactly representable floating point numbers. When performing the transformation (1) in floating point, the result is some nearby polynomial $\widetilde{Q}$. The sensitivity of a k-fold root is roughly $\varepsilon^{1/k}$ for ε denoting the relative rounding error unit (for details see the discussion following Theorem 3.1). For computation in IEEE 754 double precision ($\varepsilon = 2^{-52}$) and a 4-fold root this implies a sensitivity of about 10^{-4}. In other words, just the transformation of P into $\widetilde{Q}$ will alter the roots in the fourth decimal place.

A convenient way to estimate the effects of rounding errors is interval arithmetic. The first extensive discussion with various applications is due to Sunaga [34]. For complex numbers a circular arithmetic is appropriate as described by Gargantini and Henrici [12]. Mathematical properties of interval arithmetic can be found in many text books, among them [22, 1, 25]. A convenient implementation as a Matlab [20] toolbox is INTLAB [30].

We assume the reader to be familiar with the basic principles of interval arithmetic. In case of (1) we may carry out the transformation using interval operations yielding an interval polynomial $\mathbf{Q}$ (a polynomial with interval coefficients) with $Q \in \mathbf{Q}$. A standard way of reasoning uses inclusion monotonicity of interval arithmetic: subsequent computations and results with $\mathbf{Q}$ using interval arithmetic are true *for all* $\widetilde{Q} \in \mathbf{Q}$, among them Q as in (1).

Computations using interval arithmetic may suffer from overestimation, data dependencies and the so called wrapping effect (for details, see [25, Chapter 1.4]). A good numerical method may show poor performance when applied with interval arithmetic. This is the reason why well known methods like Sturm-sequences, Uspensky-Vincent algorithm or application of the Schur-Cohn Theorem with exact root counting by Henrici [15] had to be excluded from our investigation: accumulation of round-off and dependencies ruin the result.

We implemented all methods below in a *completely rigorous way*, that is bounding *all* effects of rounding errors by interval arithmetic. This implies that all results are mathematically correct.

The first three methods to be presented in Section 2 are based on the estimation of the difference of the roots of Q and $\overline{Q}$ and deliver a disc D containing *at least* k roots of P. Except in extraordinary circumstances these methods always deliver a result – although sometimes of poor quality.

A second class of methods compute a disc containing *exactly* k roots of P. They are based on a modification of Gershgorin circles or on Rouché's theorem. We present four such methods in Section 3.

Yet another class of methods are so-called self-validating methods. Some of those are zero finding procedures where, after transformation into fixed point form, Brouwer's fixed point theorem yields a more sophisticated sufficient criterion for a set to contain at least k roots of P. Two such methods are presented in Section 4, where the first computes directly bounds for the roots of P and the second computes bounds for eigenvalues of a comparison matrix.

A number of further methods are also mentioned in Sections 3 to 5 but numerical results are not shown for various reasons.

Extensive computational results for all methods under various circumstances are presented in Section 5. Based on the results, we present a hybrid method in the last section combining advantages of five of the preceding methods. Input to this hybrid procedure is just P and $\widetilde{z}$; the number k of roots of a nearby root cluster is determined by the method. As for all other methods the implementation takes into account all possible procedural, numerical and rounding errors such that the computed results are always mathematically correct.

We demonstrate that i) for all our examples the results of the hybrid method are almost always superior to all other methods and ii) the quality of the radius of the enclosing disc is of the order of the numerical sensitivity of the root cluster. In this sense the bounds are almost optimal.

Much of the following was inspired by discussions with my Ph.D student Prashant Batra and with Arnold Neumaier. Especially part of the collection of results is taken from Batra's Ph.D. thesis [2].

2. Perturbation bounds. To apply perturbation bounds to Q and $\overline{Q}$ as defined in (1) and (2), respectively, we first need an expansion point c. A suitable value is the mean of k zeros of P near $\widetilde{z}$. Approximations z_ν calculated by a numerical algorithm tend to lie on a circle, so that the center is a good expansion point c.

```
zeta = roots(P);
[delta,index] = sort(abs(zeta-zs));
c = mean(zeta(index(1:k)));
```

ALGORITHM 2.1. *Calculation of* `c`.

We use Matlab notation [20], where `zs` denotes $\widetilde{z}$ and `roots` is the Matlab built-in routine for computing an n-array of approximations of the roots of P. For this computed `c` let Q and $\overline{Q}$ be as in (1) and (2), respectively. Algorithm 2.1 is executable Matlab (and INTLAB) code.

A celebrated theorem by Ostrowski [26] estimates the minimum Hausdorff distance between the roots of two polynomials.

THEOREM 2.2 (Ostrowski). *Let $A(z), B(z) \in \mathbf{C}[z]$ with*

$$\begin{aligned} A(z) &= z^n + a_1 z^{n-1} + \ldots + a_n = \prod_{\nu=1}^{n} (z - \alpha_\nu) \quad \textit{and} \\ B(z) &= z^n + b_1 z^{n-1} + \ldots + b_n = \prod_{\nu=1}^{n} (z - \beta_\nu) \end{aligned}$$

be given and define

$$\gamma := 2 \cdot \max_{1 \le \nu \le n} (|a_\nu|^{1/\nu}, |b_\nu|^{1/\nu}).$$

Then the roots of A and B can be enumerated in $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_n$ respectively in such a way that

$$\max_\nu |\alpha_\nu - \beta_\nu| \le \varphi \cdot \left\{ \sum_{\nu=1}^{n} |a_\nu - b_\nu| \gamma^{n-\nu} \right\}^{1/n}$$

with $\varphi := 2n - 1$.

The result can be directly applied to Q and $\overline{Q}$ yielding a disc $D(0, \varrho)$ with

$$\varrho := \varphi \cdot \left(\sum_{\nu=0}^{k-1} |q_\nu| \gamma^\nu \right)^{1/n} \quad \text{and} \quad \gamma := 2 \max_{0 \le \nu \le k-1} |q_\nu|^{1/(n-\nu)} \tag{3}$$

containing at least k zeros of Q, showing that $D(c, \varrho)$ contains at least k zeros of P. Improvements of this result are known, the most remarkable one in [4] showing that φ can be replaced by a constant less than 4, namely $\varphi := 4 \cdot 2^{-1/n}$. The proof makes ingenious use of a property of Chebyshev polynomials established by Schönhage [32] and independently by Phillips [27].

LEMMA 2.3. *Let Γ be a continuous curve in the complex plane with end points a and b. Let $\lambda_1, \ldots, \lambda_n$ be any given points in the plane. Then there exists a point λ on Γ such that*

$$\prod_{\nu=1}^{n} |\lambda - \lambda_\nu| \geq \frac{|b-a|^n}{2^{2n-1}}.$$

Using these ingredients and adapting proofs in [27] and [4] to our special situation yields the following first bound.

THEOREM 2.4. *Let $P \in \mathbf{C}[z]$, $c \in \mathbf{C}$ and some $k \in \{1, \ldots, n\}$ be given and define $Q := P(c+z) = \sum_{\nu=0}^{n} q_\nu z^\nu$. Let R be the nonnegative root of $U(z) := z^n - 2^{2n-1} \sum_{\nu=0}^{k-1} |q_\nu| z^\nu$. Then the disc $D(c, R)$ contains at least k zeros of P.*

PROOF. Define $\overline{Q} := \sum_{\nu=k}^{n} q_\nu z^\nu$ and

$$S_t := t\overline{Q} + (1-t)Q \quad \text{for} \quad t \in [0,1], \tag{4}$$

and let $\Omega := \{z : \ S_t(z) = 0 \text{ for some } t, \ t \in [0,1]\}$. A familiar homotopy argument shows that every connected component of Ω contains as many roots of Q as of $\overline{Q}$. Denote the roots of $\overline{Q}$ by $\beta_1, \ldots, \beta_n$ with $\beta_1 = \ldots = \beta_k = 0$, and let $\overline{\Omega}$ be the union of all connected components of Ω containing some β_ν for $1 \leq \nu \leq k$. Then $\overline{\Omega}$ contains at least k roots of Q, and the roots $\alpha_1, \ldots, \alpha_n$ of Q can be enumerated such that $\alpha_\nu \in \overline{\Omega}$ for all $\nu \in \{1, \ldots, k\}$. Define $r := \max\{|z| : \ z \in \overline{\Omega}\}$. Then $|\alpha_\nu| \leq r$ for $1 \leq \nu \leq k$ and, since $\overline{\Omega}$ is closed, there is $\omega \in \overline{\Omega}$ with $|\omega| = r$. Applying Lemma 2.3 to $a = 0 \in \overline{\Omega}$ and $b = \omega \in \overline{\Omega}$, there exists $\lambda \in \overline{\Omega}$ with

$$|Q(\lambda)| = \prod_{\nu=1}^{n} |\lambda - \alpha_\nu| \geq \frac{r^n}{2^{2n-1}}. \tag{5}$$

Now $\lambda \in \overline{\Omega}$ implies $S_t(\lambda) = 0$ for some $t \in [0,1]$, and therefore by (4),

$$|Q(\lambda)| = |t(Q(\lambda) - \overline{Q}(\lambda))| \leq \sum_{\nu=0}^{k-1} |q_\nu| r^\nu,$$

so that with (5)

$$r^n - 2^{2n-1} \sum_{\nu=0}^{k-1} |q_\nu| r^\nu = U(r) \leq 0.$$

The nonnegative root R of U is a well known root bound of the Cauchy polynomial U, so that $|\alpha_\nu| \leq r \leq R$ for $1 \leq \nu \leq k$. The result follows. ∎

The value R can easily be approximated and estimated from above by some Newton iterations starting at some root bound of U. We use the Fujiwara root bound [19]

$$F(P) := 2 \cdot \max\left\{ |\frac{p_{n-1}}{p_n}|, |\frac{p_{n-2}}{p_n}|^{1/2}, \ldots, |\frac{p_1}{p_n}|^{1/(n-1)}, |\frac{p_0}{2p_n}|^{1/n} \right\} \tag{6}$$

where $P = \sum_{\nu=0}^{n} p_\nu z^\nu$, for which one can show (see [36]) that $r \leq F(P) \leq 2r$ for r denoting the nonnegative root of the Cauchy polynomial $|p_n| z^n - \sum_{\nu=0}^{n-1} |p_\nu| z^\nu$. Therefore $R \leq F(U) \leq 2R$ so that $F(U)$ is always of good quality.

Method 1. *Calculate c by Algorithm 2.1. For given k calculate Q by (1) and an upper bound ϱ for the nonnegative root of U as defined in Theorem 2.4 by some Newton iterations starting at the Fujiwara root bound $F(U)$ as in (6). Then $D(c, \varrho)$ contains at least k zeros of P. The computational effort is approximately $3kn + 4km$ operations, where m denotes the number of Newton iterations.*

The method using Theorem 2.4, which is adapted to the special situation, delivers bounds which are better by about a factor n to $2n$ than Ostrowski's bound and better by about a factor 2 to 4 than the improved bound in [4].

A rigorous implementation using interval arithmetic is straightforward. After calculating an inclusion $\mathbf{Q}$ of Q and, by formal differentiation, $\mathbf{Q}'$ (which contains Q'), let $\mathbf{r}$ be a result of a Newton iteration, all computed in interval arithmetic. Then $\varrho := \sup(\mathbf{r})$ is an upper bound for R and therefore a valid radius. A better and faster possibility is to iterate $\widetilde{r} := r - Q(r)/Q'(r)$ in ordinary floating point arithmetic and check $U(\widetilde{r}) \geq 0$ in interval arithmetic.

A drawback of the discussed perturbation bounds is that general perturbations of the coefficients of Q are taken into account and not much use is made of the fact that there is a k-fold root cluster near zero. As we will seen in the numerical tests, this is definitely necessary to obtain reasonable bounds.

The next method is based on the fact that for given P and $\widetilde{z}$ there exists z with

$$P(z) = 0 \quad \text{and} \quad |z - \widetilde{z}| \leq |P(\widetilde{z})|^{1/n}. \tag{7}$$

This observation was generalized by Montel to the case of k zeros [15, Theorem 6.4].

THEOREM 2.5 (Montel). *Let $P(z) = \sum_{\nu=0}^{n} p_\nu z^\nu$ and $Q = P(c+z) = \sum_{\nu=0}^{n} q_\nu z^\nu$ with $p_n = q_n = 1$ be given. Then for R denoting the nonnegative root of*

$$z^n - \binom{n-k}{0}|q_{k-1}|z^{k-1} - \ldots - \binom{n-2}{k-2}|q_1|z - \binom{n-1}{k-1}|q_0| \tag{8}$$

the disc $D(c,R)$ contains at least k roots of P.

Another method is based on the fact that for given P and $\widetilde{z}$ with $P'(\widetilde{z}) \neq 0$ the maximum distance of $\widetilde{z}$ to a root of P is at most n times the Newton correction:

$$\text{For } P'(\widetilde{z}) \neq 0 \text{ there exists } z \text{ with } P(z) = 0 \text{ and } |z - \widetilde{z}| \leq n \left| \frac{P(\widetilde{z})}{P'(\widetilde{z})} \right|. \tag{9}$$

This estimation is sharp [19, Theorem 33.3]; it was generalized to k zeros by van Vleck [15, Theorem 6.4].

THEOREM 2.6 (van Vleck). *Let $P(z) = \sum_{\nu=0}^{n} p_\nu z^\nu$ and $Q = P(c+z) = \sum_{\nu=0}^{n} q_\nu z^\nu$ with $p_n = q_n = 1$ be given. Assume $q_k \neq 0$ and denote by R the nonnegative root of*

$$|q_k|z^k - \binom{n-k+1}{1}|q_{k-1}|z^{k-1} - \ldots - \binom{n-1}{k-1}|q_1|z - \binom{n}{k}|q_0|. \tag{10}$$

Then the disc $D(c,R)$ contains at least k roots of P.

Note that both theorems (as Method 1) provide a direct bound for the radius of a disc containing k roots of P. An upper bound ϱ for R can be computed by some Newton iterations as before. As we will see later in the computational results, van Vleck's bound is generally superior to Montel's because the root bound of the former depends basically the k-th root rather the n-th root of certain quantities.

Methods 2 and 3. *Calculate c by Algorithm 2.1. For given k calculate Q by (1) and an upper bound ϱ for R according to Theorems 2.5 and 2.6 by some Newton iterations starting at the Fujiwara root bound for (8) and (10), respectively. Then $D(c,\varrho)$ contains at least k roots of P. The computational effort is approximately $3kn + 4km$ operations, where m denotes the number of Newton iterations.*

An implementation delivering rigorous bounds is again straightforward. The three methods presented in this section compute directly a disc $D(c,\varrho)$ containing *at least* k roots of P and – except under extraordinary numerical circumstances – never fail, although the bound may be poor. In the following section we present methods and sufficient criterions for a computable disc $D(c,\varrho)$ containing *exactly* k roots of P.

3. Discs containing exactly k roots. The criterions so far need to calculate the shifted polynomial Q. For higher degrees n this bears the disadvantage that large binomial coefficients $\binom{n}{\nu}$ are involved which may cause significant round-off and cancellation errors. The next method works with the original polynomial P and Durand-Kerner corrections. The latter are usually more stable to calculate, especially for larger n. For a given polynomial $P = \sum_{\nu=0}^{n} p_\nu z^\nu$, $p_n = 1$ the set of eigenvalues of the companion matrix

$$A = \begin{pmatrix} 0 & & & -p_0 \\ 1 & 0 & & -p_1 \\ & \ddots & & \vdots \\ & & 1 & -p_{n-1} \end{pmatrix} \tag{11}$$

is the same as the set of roots of P. One may apply Gershgorin's theorem to A to obtain well known but crude bounds for the roots of P, cf. [3] and [38]. Generalizations of the companion matrix are known to improve those bounds, cf. [33], [5], [6], [10], [8] and [7]. A new result in this direction is the following [23].

Let $P(z) = \sum_{\nu=0}^{n} p_\nu z^\nu$, $p_n \neq 0$, be a polynomial with roots $\zeta_1, \ldots, \zeta_n$ so that

$$P(z) = p_n \prod_{\nu=1}^{n} (z - \zeta_\nu).$$

Let $z_1, \ldots, z_n$ be pairwise distinct approximations to $\zeta_1, \ldots, \zeta_n$ and

$$T(z) := \prod_{\nu=1}^{n} (z - z_\nu).$$

Then the partial fraction expansion of P/T has the form

$$\frac{P(z)}{T(z)} = p_n + \sum_{\nu=1}^{n} \frac{\alpha_\nu}{z - z_\nu},$$

where the coefficients α_ν can be identified as

$$\alpha_\nu = \frac{P(z_\nu)}{\prod_{\mu \neq \nu} (z_\nu - z_\mu)}. \tag{12}$$

The quantities α_ν / p_n are the Durand-Kerner [9, 17] corrections to the approximations z_ν, which apparently go back to Weierstraß [37]. For simple roots they define the quadratically convergent Durand-Kerner iteration. With these notations, Neumaier [23] showed the following Gershgorin-like result.

THEOREM 3.1 (Neumaier). *If $p_n \neq 0$ then all roots of P belong to the union $\mathfrak{S}$ of the discs*

$$D_\nu := D(z_\nu - r_\nu, |r_\nu|) \text{ with } r_\nu := \frac{n}{2} \cdot \frac{\alpha_\nu}{p_n}. \tag{13}$$

Moreover, every connected component of $\mathfrak{S}$ consisting of m of these discs contains exactly m zeros of $P(z)$, counting them with their algebraic multiplicity.

The assumption that the approximations z_ν are pairwise distinct may appear as an obstacle for the application of Neumaier's theorem. This is not the case. Given a k-fold root $\widehat{z}$ of P, a numerical algorithm generally computes approximations of the form $z_\nu = \widehat{z} + \sigma e^{2\pi i \cdot \nu / k}$ for $\nu = 1, \ldots, k$, where σ is of the order of the numerical sensitivity of $\widehat{z}$. This sensitivity with respect to ε-perturbations in the coefficients p_ν is well known [39, Section 7.4] to be

$$\sigma \approx \left(\varepsilon \frac{|P|(|\widehat{z}|)}{|P^{(k)}(\widehat{z})/k!|} \right)^{1/k}, \tag{14}$$

where $|P| = \sum_{\nu=0}^{n} |p_\nu| z^\nu$. So the next method is an example where approximations should not be "too" good. Indeed, in case of existence of multiple roots ζ_ν the exact values would not work as approximations z_ν. Fortunately, numerical inaccuracies do us the favor to produce approximations with the desired properties.

Method 4. *Based on approximations z_ν, $1 \le \nu \le n$, to the roots of P computed by some numerical routine, calculate the quantities α_ν by (12) and the discs D_ν by Theorem 3.1. Compute the number m of discs D_ν belonging to a connected component near $\widetilde{z}$ and an enclosing disc D for those discs. Then D contains exactly m roots of P. Given approximations z_ν, the method requires approximately $5n^2$ operations.*

As will be seen in the numerical results this method is advantageous for larger degrees because it works with the original polynomial P rather than with a shifted polynomial Q. However, the radii in (13) grow with a factor n rather than with k, which may be a major drawback. But Theorem 3.1 gives the possibility to identify connected components of circles containing exactly k roots so that those k roots can be separated from the others. And based on that information Neumaier gave a possibility to refine the enclosing discs. where the radii of the refined discs grows with about a factor k. The refinement is possible if the connected components are well enough separated.

THEOREM 3.2 (Neumaier). *With the notations of Theorem 3.1 denote by C the set of indices of a connected component of $\mathfrak{S}$. Let real λ be given such that*

$$\lambda \le \sum_{\mu \notin C} Re \frac{\alpha_\mu}{p_n(z - z_\mu)} \qquad \text{for all } z \in D_\nu, \nu \in C$$

and assume

$$\beta := 1 + \lambda > \frac{|C|}{n}.$$

Then the roots of P belonging to the cluster D_ν, $\nu \in C$ are also contained in the union of the discs

$$D'_\nu := D(z_\nu - r'_\nu, |r'_\nu|) \text{ with } r'_\nu := \frac{|C|}{2\beta} \cdot \frac{\alpha_\nu}{p_n}. \tag{15}$$

In a practical application the quantities λ are usually small and therefore β near 1. That means, if the connected component C is separated by some factor $n/2$, then the radii of the improved inclusions shrink this factor from $n/2$ to the order of $k/2$ times the Durand-Kerner corrections α_ν/p_n, where the quality of the latter depends on the quality of the approximations z_ν. A lower bound λ can easily be computed from the given discs D_ν using interval arithmetic. The improvement can be quite significant (see the computational results in Section 5) at small additional costs. This defines our next method.

Method 5. *Based on a connected component of m enclosing discs for m roots of P computed by Method 4, compute improved enclosing discs according to Theorem 3.2. The method requires some additional $2n$ operations.*

The next method is, like many others, based on Rouché's theorem. Basically this says that if $|f(z) - g(z)| < |g(z)|$ on the boundary of $D(c, r)$ for some holomorphic functions f, g, then f and g have the same number of roots in the interior of D. The proof is a simple application of the principle of argument. A corollary is the following result due to Pellet [19].

THEOREM 3.3 (Pellet). *Let $Q = \sum_{\nu=0}^{n} q_\nu z^\nu$ with $q_n q_0 \neq 0$ be given. If the polynomial*

$$V(z) := \sum_{\nu=k+1}^{n} |q_\nu| z^\nu - |q_k| z^k + \sum_{\nu=0}^{k-1} |q_\nu| z^\nu \tag{16}$$

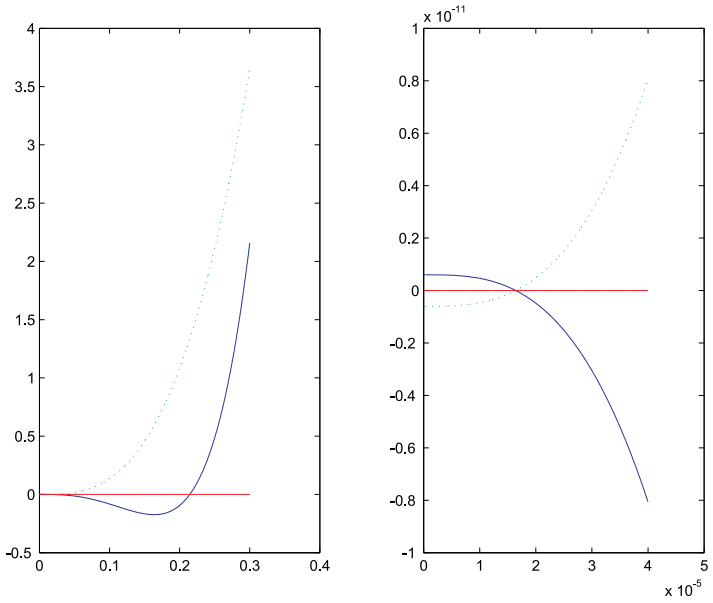
has two positive roots r and R with $r < R$, then Q has exactly k zeros in $D(0, r)$ and no zeros in $\{z : r < |z| < R\}$.

The criterion makes nearly optimal use of the used information, namely the absolute values of the coefficients of P (cf. Walsh [19]). Pellet's theorem can be applied directly to the shifted polynomial Q as in (1), such that $D(c,r)$ contains exactly k zeros of P.

The problem remains to approximate the smaller nonnegative root r of V as in (16). The coefficients $|q_\nu|$, $0 \le \nu \le k-1$ are small compared to $|q_k|$ if the k roots of P near c are well enough separated from the others. If V has two nonnegative roots r, R and the diameter of the root cluster is not too large, then r is small enough so that the terms $|q_\nu| r^\nu$ are also small compared to $|q_k| r^k$ for $k+1 \le \nu \le n$. That means the nonnegative root $\widetilde{r}$ of

$$W(z) := |q_k| z^k - \sum_{\nu=0}^{k-1} |q_\nu| z^\nu \tag{17}$$

is usually a good approximation to r. Therefore, as in van Vleck's bound, Pellet's bound is proportional to the k-th rather than the n-th root of certain quantities. A typical graph of V and W for a polynomial of degree 10 with a three-fold root looks as follows.



GRAPH 3.4. *Behavior of V (solid) and W (dotted)*

Note the magnitude of the axes; the right graph is zoomed into the interesting region near the smaller root of V. By the construction, the nonnegative root $\widetilde{r}$ of W is always a lower bound for (a possibly existing) smallest nonnegative root r of V. This suggests the following sixth method.

Method 6. *Calculate c by Algorithm 2.1. For given k calculate Q by (1) and V and W by (16) and (17), respectively. Then proceed as follows:*

- *i) Starting at the Fujiwara root bound for W perform some Newton iterations to obtain an approximation $\widetilde{r}$ for the nonnegative root of W.*
- *ii) Set $r := \widetilde{r} - V(\widetilde{r})/V'(\widetilde{r})$.*
- *iii) If $V(r) > 0$, increase r stepwise until $V(r) \le 0$. If not successful, return "no bound".*
- *iv) Perform some Newton iterations on V until r sufficiently accurate.*

If successful, $D(c,r)$ contains exactly k roots of P. The computational effort is approximately $3n^2 + 4k \cdot m_1 + 4n \cdot m_2$, where m_1 denotes the number of iterations in step i) and m_2 denotes the total number of iterations in steps ii), iii) and iv).

Note that Method 6 offers the additional advantage of computing the distance to the remaining roots of P by inspecting R as in Theorem 3.3. It will turn out that Method 6 delivers, in many cases, very good results so that we choose to use it in our hybrid method – with an improved method to determine the radius r.

The previous three methods deliver the exact number of roots within a certain disc, where Method 4 – except extraordinary circumstances – *always* delivers an answer, possibly with $l > k$ roots, but Method 6 either gives a disc with exactly k roots or no answer at all. There are more methods of the latter type in the form of a sufficient criterion that $D(c,r)$ contains exactly k roots of P. The first we mention is a general result by Neumaier [24] for holomorphic functions.

THEOREM 3.5 (Neumaier). *Let f be holomorphic in the interior of $D(c,r)$ and let ε with $0 < \varepsilon < r$ be given. If*

$$\left| Re \frac{f^{(k)}(z)}{k!} \right| > \sum_{\nu=0}^{k-1} \left| \frac{f^{(\nu)}(c)}{\nu!} \right| \varepsilon^{\nu-k} \text{ for all } z \in D(c,r), \tag{18}$$

then f has exactly k roots in $D(c,\varepsilon)$ counting multiplicities.

The theorem has been adapted by Batra to our problem of polynomial root bounding as follows [2, Theorem 2.6.3].

THEOREM 3.6 (Batra). *Let Q be as in (1), $q_n \neq 0$, and $r > 0$ be given. If*

$$\varphi_z(r) := r^k \cdot \left| Re \sum_{\nu=k}^{n} q_\nu z^{\nu-k} \right| - \sum_{\nu=0}^{k-1} |q_\nu| r^\nu > 0 \tag{19}$$

for all z on the boundary of $D(0,r)$, then $D(c,r)$ contains exactly k roots of P counting multiplicities.

Batra's criterion was, in our computational tests, always superior to Neumaier's because it takes advantage of the special structure of the functions in use.

At first sight such a criterion seems difficult to apply since (19) has to be verified for *all* $z \in \partial D(0,r)$. This is not the case. A straightforward approach using circular arithmetic as defined in [12] with z replaced by $Z := D(0,r)$ and suitable control of rounding errors delivers a disc which includes

$$\left\{ \sum_{\nu=k}^{n} q_\nu z^{\nu-k} : \ z \in Z \right\}.$$

With this it is easy to verify (19). Following is the *executable* code in INTLAB [30], the Matlab interval toolbox, to demonstrate the simplicity of computation:

```
Z = midrad(0,r); R = intval(r);
t1 = Q(n); t2 = abs(Q(k-1));
for nu = n-1:-1:k
    t1 = t1 * Z + Q(nu);
end
for nu = k-2:-1:0
    t2 = t2 * R + abs(Q(nu));
end
success = ( inf(R^k * abs(real(t1))) > sup(t2) );
```

ALGORITHM 3.7. *Verification of (19)*

Concerning rigorous bounds we mention that some other criterions are especially easy to verify with rigor. For example, $V(r) \leq 0$ for V as in (16) and some $r > 0$ implies that $D(c,r)$ contains exactly k roots of P. If V is evaluated at r in floating point arithmetic using Horner's scheme and rounding upwards (cf. [16]), then the computed result $\widetilde{y}$ is greater or equal to the true result $y := V(r)$, such that $\widetilde{y} \leq 0$ implies that the assertions are valid.

It turns out, however, that using circular arithmetic Batra's criterion cannot be superior to Pellet's. For

$Z := D(0,r)$ one computes

$$t_1 = \sum_{\nu=k}^{n} q_\nu Z^{\nu-k} = D(q_k, \sum_{\nu=k+1}^{n} |q_\nu| r^{\nu-k}) \tag{20}$$

if the left hand side of (20) is evaluated in circular arithmetic [12]. It follows that $r^k|\mathrm{Re}\, t_1| - t_2 > 0$ implies $V(r) < 0$.

But (19) is to be verified only for $z \in \partial D(0,r)$ whereas Algorithm 3.7 checks (19) for all $z \in D(0,r)$. Therefore, one may try to divide the circle $\partial D(0,r)$ into segments and check (19) individually. Assume that P is a real polynomial. Then for $z = re^{i\alpha}$, $\mathrm{Re} \sum_{\nu=k}^{n} q_\nu z^{\nu-k} = \sum_{\nu=k}^{n} q_\nu r^{\nu-k} \cos((\nu-k)\alpha)$. Replacing α by small intervals covering $[0,\pi]$ may yield an improvement. However, in all our examples we did not observe any difference at all. Moreover, the evaluation of $\left|\mathrm{Re} \sum_{\nu=k}^{n} q_\nu Z^{\nu-k}\right|$ with $Z = D(0,r)$ in (circular) interval arithmetic was always almost equal to the set $\left\{\left|\mathrm{Re} \sum_{\nu=k}^{n} q_\nu z^{\nu-k}\right| : \ z \in \partial D(0,r)\right\}$ in our numerical tests.

Although Pellet's criterion makes nearly optimal use of the available information, an improvement, at least for real polynomials, has been given by Batra [2].

THEOREM 3.8 (Batra). *Let a real polynomial $Q = \sum_{\nu=0}^{n} q_\nu z^\nu$ with $q_n \neq 0$ be given. If*

$$\varphi(r) := |q_k r^k| - \sum_{\nu=1}^{n} |q_{k-\nu} r^{k-\nu} + q_{k+\nu} r^{k+\nu}| > 0 \quad \textit{where} \quad q_\nu := 0 \textit{ for } \nu \notin \{0,\ldots,n\},$$

then Q has exactly k roots in the open disc $D(0,r)$.

The advantage of this criterion is that $q_{k-\nu}$ and $q_{k+\nu}$ may cancel; however, as has been discussed earlier, for small r the governing term in φ is $|q_k r^k|$, so that we hardly observed any difference to Pellet's method.

In the application of Pellet's theorem in Method 6 the radius of the enclosing disc was given by the root of the polynomial V in (16). Theorem 3.6 and Theorem 3.8 are sufficient criterions for a given radius r. Before we come to the question how to determine this radius r we mention another, new criterion by Neumaier, also based on Rouché's theorem.

THEOREM 3.9 (Neumaier). *Let $P(z) = \sum_{\nu=0}^{n} p_\nu z^\nu$, $p_n \neq 0$, some $k \in \{1,\ldots,n\}$ and pairwise distinct $z_1,\ldots,z_n \in \mathbb{C}$ be given. If for the constants α_ν as in (12) and some given $c \in \mathbb{C}$, $0 < r \in \mathbb{R}$,*

$$\varphi(r) := Re\left(p_n + \sum_{\nu=1}^{n} \frac{(c-z_\nu)^* \alpha_\nu}{|c-z_\nu|^2 - r^2}\right) - r \sum_{\nu=1}^{n} \left|\frac{\alpha_\nu}{|c-z_\nu|^2 - r^2}\right| > 0,$$

then the number of roots of P in $D(c,r)$ is exactly equal to the number of z_ν in $D(c,r)$.

This result offers the advantage that, after pre-computation of the constants α_ν, the midpoint c of the potential inclusion disc $D(c,r)$ may be varied without much additional cost. Also note that this method, too, works with the original data P and the Durand-Kerner corrections α_ν/p_n rather than with the shifted polynomial Q. This is advantageous for larger degrees n.

Since in our experience approximations of the roots calculated by standard numerical algorithms are very accurate with errors of the order of the numerical sensitivity σ as in (14), we used in the application of Theorems 3.8 and 3.9 the center c as produced by Algorithm 2.1.

Theorem 3.9 presents a sufficient criterion for a *given* radius r, where other methods bear the advantage to *determine* the radius of the inclusion. Usually, the accuracy of the radius itself is not so important, two or three digits may suffice. Therefore we determine r by a simple bisection strategy.

i) Calculate σ according to (14), if $\varphi(\sigma) \leq 0$ goto iii).

ii) [r_2 valid radius] Set $r_2 := \sigma$ and decrease σ to r_1 until $\varphi(r_1) \le 0$;
goto iv).
iii) [r_1 no valid radius] Set $r_1 := \sigma$ and increase σ to r_2 until $\varphi(r_2) > 0$.
If not successful, return no bound.
iv) [r_1 not valid, r_2 valid radius] Bisect $[r_1, r_2]$ subject to $\varphi > 0$ until accurate enough.

ALGORITHM 3.10. *Bisection for r based on φ*

With this we can formulate our seventh method.

Method 7. *Based on approximations z_ν, $1 \le \nu \le n$, to the roots of P computed by some numerical routine, calculate the quantities α_ν by (12). Determine r by Algorithm 3.10 with φ as in Theorem 3.9. If successful, $D(c, r)$ contains exactly k roots of P. Given approximations z_ν, the computational effort is approximately $3n^2 + 4mn$ operations, where m denotes the number of evaluations of φ.*

4. Self-validating methods. As mentioned before one principle of self-validating methods is the transformation of a root-finding problem into a fixed point problem and application of Brouwer's fixed point theorem. As an example consider a polynomial P with a simple root near some $\widetilde{z}$. Let $R \neq 0$ be some complex number, $Z := D(\widetilde{z}, \varepsilon)$ and for some $\varepsilon > 0$ assume

$$K(Z) := \widetilde{z} - R \cdot P(\widetilde{z}) + \{1 - R \cdot P'(Z)\}(Z - \widetilde{z}) \subseteq \mathrm{int}(Z), \tag{21}$$

where $\mathrm{int}(Z)$ denotes the interior of Z. For the moment all operations are assumed to be power set operations, and $P'(Z) := \{P'(z) : z \in Z\}$. The operator K is the so called Krawczyk operator introduced for the computation of error bounds for zeros of systems of nonlinear equations [18]. A standard argument shows that (21) implies that there exists exactly one root of P in Z: Define $f(z) := z - R \cdot P(z)$, then for every $z \in Z$ there exists $\xi_z \in Z$ with $f(z) = f(\widetilde{z}) + f'(\xi_z)(z - \widetilde{z}) = \widetilde{z} - R \cdot P(\widetilde{z}) + (1 - R \cdot P'(\xi_z))(z - \widetilde{z}) \in K(Z) \subseteq Z$, which means that the continuous function f maps the nonempty, closed and convex set Z into itself. Brouwer's fixed point theorem implies existence of $\widehat{z} \in Z$ with $f(\widehat{z}) = \widehat{z} = \widehat{z} - R \cdot P(\widehat{z})$ and therefore $P(\widehat{z}) = 0$ by $R \neq 0$. Uniqueness of $\widehat{z} \in Z$ with $P(\widehat{z}) = 0$ follows because $P'(z) = 0$ for $z \in Z$ implies $\{z - R \cdot P(\widetilde{z}) : z \in Z\} \subseteq K(Z) \subseteq \mathrm{int}(Z)$, a contradiction because the shifted set Z cannot be contained in its interior.

The method is applicable to systems of nonlinear equations $F(z) = 0$ as well [18], and there are various improvements [25, 29]. For an application on digital computers the left hand side of (21) is evaluated using interval arithmetic, where the Jacobian $\partial F / \partial x(Z)$ can be evaluated using automatic differentiation [28, 25].

This kind of self-validating methods verifies uniqueness of a zero in Z and is therefore, by principle, not suited for the computation of bounds for multiple zeros or root clusters. In the case of polynomials we may create a self-validating method as follows. Let $A(z) := \sum_{\nu=0}^{k} a_\nu z^\nu$ and $B(z) := \sum_{\nu=0}^{n-k} b_\nu z^\nu$. By fixing $a_k := 1$ and $b_{n-k} := p_n$,

$$P - A \cdot B = 0 \tag{22}$$

defines, by comparing coefficients, a system of n equations in the n unknowns $a_0, \ldots, a_{k-1}, b_0, \ldots, b_{n-k-1}$. This nonlinear system can be solved by a self-validating method, for example by the function `verifynlss` in INTLAB [30], which is based on the method as described above with various improvements as in [29]. This yields inclusions $\mathbf{a}_\nu$, $0 \le \nu \le k - 1$, of coefficients of a polynomial A being a factor of P. A root bound r of $\mathbf{A}(c + z)$, which is by definition also a root bound of the shifted factor $A(c + z)$, yields a disc $D(c, r)$ containing at least k zeros of P. This defines our eighth method. The computational effort is $\mathcal{O}(n^3)$ operations, so in terms of computing time this method is not competitive with previous ones.

Method 8. *Compute inclusion of coefficients of factor A as in (22) and proceed as described before.*

One may apply the same idea to Q such that the factor need not be shifted. Due to accumulation of rounding errors this gives much worse results.

Another self-validating method yielding a nonlinear system in only k unknowns can be derived as follows. Define $A(z) := \sum_{\nu=0}^{k} a_\nu z^\nu$, $a_k = 1$, where the a_ν, $0 \le \nu \le k-1$, are the unknowns. Then perform a formal polynomial division P/A using interval arithmetic and automatic differentiation. The result is a remainder term $R(z)$ of degree less than k. Setting the k coefficients of R to zero defines a nonlinear system of k equations in the k unknowns $a_0 \dots a_{k-1}$. We tested this approach but due to accumulated dependencies the results were mostly poorer than those the previous method; we therefore discarded this approach.

Finally, there is a recent paper [**?**] on the inclusion of multiple or nearly multiple eigenvalues of a matrix. This self-validating method can directly be applied to the companion matrix A as in (11) and is available as INTLAB-routine `verifyeig` [30]. The method requires approximations for an invariant subspace to an eigenvalue λ. Such a basis can be computed from Horner's scheme. For λ being a k-fold eigenvalue of A,

$$x = \begin{pmatrix} \lambda^{n-1} + p_{n-2}\lambda^{n-2} + \ldots + p_1 \\ \cdots \\ \lambda + p_{n-1} \\ 1 \end{pmatrix} \tag{23}$$

is the only eigenvector to λ because the geometric multiplicity of λ is always one. Another argument for this fact is the sensitivity: The polynomial coefficients are the input to the polynomial root and to the matrix eigenvalue problem, so that comparing the sensitivities gives the result. The first component of x times λ is equal to $-p_0$ by definition of λ. Principal vectors can be computed by successive formal differentiation of the components of x with respect to λ and inserting the value of λ. This is our ninth method. The result is a complex disc containing at least k eigenvalues of A which means k roots of P; the computational effort is again $\mathcal{O}(n^3)$.

Method 9. *Compute inclusion of eigenvalues of the comparison matrix A by the INTLAB-routine* `verifyeig` *[30] based on [**?**] using approximations for the invariant subspace as in (23).*

5. Computational results. In the following we present extensive numerical tests. The tests and their interpretation refer to the inclusion of one root cluster; for the computation of several clusters or all roots of a polynomial other methods may be more advantageous. A first set of test polynomials of degree n with a k-fold root is generated by

$$P = (x - \widetilde{z})^k \left(x^{n-k} + \sum_{\nu=0}^{n-k-1} r_\nu x^\nu \right) \qquad \text{for } \widetilde{z} = 2, \tag{24}$$

where r_ν are uniformly distributed random numbers in $[-1, 1]$ with "few leading nonzero digits", that is with a binary expansion $\pm \sum_{\mu=0}^{25} b_\mu \cdot 2^{1-\mu}$. This assures that i) the coefficients of P are exactly representable in IEEE 754 double precision, that ii) the computed P has an exact k-fold root $\widetilde{z}$ with iii) of the order $k + \log(n-k)$ real and the rest complex roots. Note that the separation of the k-fold root $\widetilde{z}$ from the other roots is not known a priori; accidentally the distance might be very small or even zero.

For such a polynomial we compute a disc $D(c, r)$ containing at least or exactly k roots by one of the methods described before. We display the median and maximum of the radii r for 100 test polynomials for every method, and in the column "failed" the number of test cases (out of 100) where the method failed. For better comparison of the quality of the root bounds we display the median and maximum ratio r/σ in the last two columns, where in a final row we display the median and maximum sensitivity σ of the root according to (14).

In addition to the nine methods described so far we also give the test results for our tenth, the hybrid method to be presented in the next section. This allows easier comparison of the results and we felt that the computational results, the reason for our new hybrid method, should be seen before the method is described. The hybrid method includes a guess of the number k of roots near the given approximation $\widetilde{z}$. Therefore we

give two lines of results: Method 10 is the hybrid Algorithm 6.4 with prespecified number of roots k, and the second last row shows the results with computed k as by the forthcoming Algorithm 6.4.

As mentioned before all results displayed below are based on an implementation taking into account all computational and rounding errors. That means the results by all algorithms are rigorous and mathematically correct.

The first set of test polynomials shows the results for fixed $k = 3$ and for $n = 20$, $n = 40$ and $n = 100$, respectively. A detailed discussion of the computational results will follow at the end of this section.

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	1.7e+0	1.8e+0	0	49748.9	64153.5
Montel	2	4.6e-1	4.7e-1	0	13366.3	17168.0
van Vleck	3	2.3e-4	3.7e-4	0	6.7	7.8
Neumaier-Gershgorin	4	3.5e-4	1.3e-2	0	9.2	394.2
Neumaier-refinement	5	6.2e-5	1.9e-3	0	1.7	58.2
Pellet	6	2.2e-5	3.5e-5	0	0.6	0.7
Neumaier-Rouché	7	7.1e-5	2.0e-3	0	1.9	63.7
factor	8	5.5e-5	1.5e-4	0	1.7	2.8
eigenvalues	9	2.7e-5	4.2e-5	0	0.8	0.9
hybrid with k	10	2.2e-5	3.5e-5	0	0.6	0.7
hybrid	Alg. 6.4	2.2e-5	3.5e-5	0	0.6	0.7
sensitivity	σ	3.3e-5	5.5e-5	0	1.0	1.0

TABLE 5.1. *Results for 100 samples (24) for $n = 20$, $k = 3$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	4.0e+0	4.1e+0	0	110762.5	152089.1
Montel	2	1.0e+0	1.0e+0	0	27858.0	38301.7
van Vleck	3	4.9e-4	9.5e-4	0	13.8	16.3
Neumaier-Gershgorin	4	6.8e-4	1.4e-2	0	17.8	310.3
Neumaier-refinement	5	5.9e-5	9.6e-4	0	1.6	21.4
Pellet	6	2.3e-5	4.4e-5	0	0.6	0.8
Neumaier-Rouché	7	6.7e-5	9.9e-4	0	1.7	22.1
factor	8	5.7e-5	2.7e-4	0	1.7	4.1
eigenvalues	9	2.7e-5	5.0e-5	0	0.8	0.9
hybrid with k	10	2.3e-5	4.4e-5	0	0.6	0.8
hybrid	Alg. 6.4	2.3e-5	4.4e-5	0	0.6	0.8
sensitivity	σ	3.6e-5	6.7e-5	0	1.0	1.0

TABLE 5.2. *Results for 100 samples (24) for $n = 40$, $k = 3$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	7.7e+1	7.8e+1	0	2.1e+6	2.9e+6
Montel	2	2.0e+1	2.0e+1	0	551957.9	754245.3
van Vleck	3	9.0e+50	2.3e+51	0	2.4e+55	4.0e+55
Neumaier-Gershgorin	4	1.7e-3	1.4e-2	0	47.1	301.8
Neumaier-refinement	5	5.8e-5	3.7e-4	0	1.6	8.0
Pellet	6	-	-	100	-	-
Neumaier-Rouché	7	6.5e-5	4.2e-4	0	1.8	9.0
factor	8	5.9e-5	1.8e-4	0	1.7	3.0
eigenvalues	9	2.8e-5	4.5e-5	0	0.8	0.9
hybrid with k	10	5.8e-5	3.7e-4	0	1.6	8.0
hybrid	Alg. 6.4	5.8e-5	3.7e-4	0	1.6	8.0
sensitivity	σ	3.6e-5	5.9e-5	0	1.0	1.0

TABLE 5.3. *Results for 100 samples (24) for $n = 100$, $k = 3$*

Most methods show a slight dependence on the degree of the polynomial. The biggest dependency is seen for method 1 and 2 because of the degree n of U in Theorem 2.4 and of the Montel polynomial in (8). Those two methods also give the poorest results. The other methods 3 to 10 give results not far or even better than the (worst case) sensitivity of the root. An exception is van Vleck's method for $n = 100$. Here, the computed q_k, in this case q_3, was almost zero in one case so that the computed bound was very poor. Such occurrences have to be taken care of in the hybrid method. Note how the refinement by Method 6 improves the bounds computed by Method 5. The bounds by bounding eigenvalues as in Method 9 are the best for $n = 100$, but at significantly higher computational costs.

The next tables show the results for $n = 40$, $k \in \{1, 2, 5\}$ and for $n = 100$, $k = \{1, 5, 20\}$, respectively. Especially the estimations for a simple root and Methods 1 and 2 are extremely poor; all other methods show results not too far from the theoretical sensitivity. Note that for larger values of n and k there are a number of failures. In all cases method 4, the Neumaier-Gershgorin bound, shows no failures and gives reasonable results. Also note the extreme sensitivity of the root for $n = 100$ and $k = 20$.

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	3.3e+0	3.4e+0	0	1.2e+15	2.6e+15
Montel	2	8.5e-1	8.7e-1	0	3.0e+14	6.6e+14
van Vleck	3	1.7e-13	1.1e-12	0	52.3	202.2
Neumaier-Gershgorin	4	1.0e-13	6.2e-13	0	31.2	108.6
Neumaier-refinement	5	3.1e-15	1.6e-14	0	0.9	3.2
Pellet	6	4.4e-15	2.7e-14	0	1.4	5.3
Neumaier-Rouché	7	4.8e-15	3.0e-14	0	1.4	5.5
factor	8	4.7e-15	2.6e-14	3	1.4	5.3
eigenvalues	9	4.7e-15	2.9e-14	0	1.4	5.2
hybrid with k	10	6.7e-16	6.7e-16	0	0.2	0.5
hybrid	Alg. 6.4	6.7e-16	1.1e-15	0	0.2	0.5
sensitivity	σ	2.8e-15	1.4e-14	0	1.0	1.0

TABLE 5.4. *Results for 100 samples (24) for $n = 40$, $k = 1$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	3.6e+0	3.7e+0	0	3.4e+7	4.8e+7
Montel	2	9.0e-1	9.2e-1	0	8.6e+6	1.2e+7
van Vleck	3	1.3e-6	2.8e-6	0	11.4	15.9
Neumaier-Gershgorin	4	2.9e-6	7.1e-6	0	26.8	63.0
Neumaier-refinement	5	1.5e-7	3.4e-7	0	1.3	2.8
Pellet	6	4.5e-8	1.0e-7	0	0.4	0.6
Neumaier-Rouché	7	1.7e-7	3.9e-7	0	1.5	3.5
factor	8	7.6e-8	3.5e-7	2	0.8	1.6
eigenvalues	9	5.7e-8	1.4e-7	0	0.5	0.8
hybrid with k	10	4.5e-8	1.0e-7	0	0.4	0.6
hybrid	Alg. 6.4	4.5e-8	1.0e-7	0	0.4	0.6
sensitivity	σ	1.0e-7	2.2e-7	0	1.0	1.0

TABLE 5.5. *Results for 100 samples (24) for $n = 40$, $k = 2$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	4.9e+0	5.0e+0	0	1338.6	1566.8
Montel	2	1.2e+0	1.2e+0	0	325.8	382.2
van Vleck	3	4.3e-2	5.9e-2	0	11.5	12.6
Neumaier-Gershgorin	4	4.9e-2	4.7e-1	0	13.2	125.0
Neumaier-refinement	5	7.0e-3	5.6e-2	0	1.8	15.1
Pellet	6	2.9e-3	4.1e-3	0	0.8	0.9
Neumaier-Rouché	7	8.0e-3	6.2e-2	0	2.2	16.8
factor	8	8.6e-3	2.4e-2	0	2.3	4.7
eigenvalues	9	3.2e-3	4.3e-3	0	0.9	0.9
hybrid with k	10	2.9e-3	4.1e-3	0	0.8	0.9
hybrid	Alg. 6.4	2.9e-3	4.1e-3	0	0.8	0.9
sensitivity	σ	3.7e-3	5.0e-3	0	1.0	1.0

TABLE 5.6. *Results for 100 samples (24) for $n = 40$, $k = 5$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	5.7e+0	5.7e+0	0	1.6e+15	5.3e+15
Montel	2	1.4e+0	1.4e+0	0	4.1e+14	1.3e+15
van Vleck	3	4.4e-16	4.4e-16	0	0.1	0.4
Neumaier-Gershgorin	4	4.0e-13	1.6e-12	0	110.7	457.2
Neumaier-refinement	5	4.7e-15	1.7e-14	0	1.2	4.9
Pellet	6	4.4e-16	8.9e-16	0	0.1	0.4
Neumaier-Rouché	7	8.4e-15	3.1e-14	0	2.1	9.3
factor	8	7.5e-15	3.0e-14	1	2.0	9.1
eigenvalues	9	7.8e-15	2.9e-14	0	2.1	9.0
hybrid with k	10	6.7e-16	6.7e-16	0	0.2	0.6
hybrid	Alg. 6.4	6.7e-16	1.1e-15	0	0.2	0.6
sensitivity	σ	3.2e-15	1.1e-14	0	1.0	1.0

TABLE 5.7. *Results for 100 samples (24) for $n = 100$, $k = 1$*

method		radius median	radius max	failed	ratio to sensitivity median	ratio to sensitivity max
Ostrowski-R.	1	7.9e+1	7.9e+1	0	20964.5	25365.0
Montel	2	2.2e+1	2.2e+1	0	5871.2	7103.6
van Vleck	3	3.5e+26	5.0e+26	0	9.2e+28	1.2e+29
Neumaier-Gershgorin	4	1.1e-1	5.4e-1	0	28.2	130.0
Neumaier-refinement	5	5.9e-3	2.3e-2	0	1.5	5.9
Pellet	6	-	-	100	-	-
Neumaier-Rouché	7	6.6e-3	2.8e-2	0	1.7	6.7
factor	8	9.0e-3	2.0e-2	0	2.5	4.3
eigenvalues	9	3.2e-3	4.2e-3	0	0.9	0.9
hybrid with k	10	5.9e-3	2.3e-2	0	1.5	5.9
hybrid	Alg. 6.4	5.9e-3	2.3e-2	0	1.5	5.9
sensitivity	σ	3.7e-3	4.9e-3	0	1.0	1.0

TABLE 5.8. *Results for 100 samples (24) for $n = 100$, $k = 5$*

method		radius median	radius max	failed	ratio to sensitivity median	ratio to sensitivity max
Ostrowski-R.	1	8.8e+1	8.9e+1	0	378.1	435.9
Montel	2	3.4e+1	3.5e+1	0	147.1	169.6
van Vleck	3	7.4e+5	1.0e+6	0	3.2e+6	4.7e+6
Neumaier-Gershgorin	4	7.8e+0	1.7e+2	0	33.5	723.7
Neumaier-refinement	5	7.8e+0	1.7e+2	0	33.5	723.7
Pellet	6	-	-	100	-	-
Neumaier-Rouché	7	-	-	100	-	-
factor	8	-	-	100	-	-
eigenvalues	9	-	-	100	-	-
hybrid with k	10	4.3e+0	1.1e+1	0	17.8	45.2
hybrid	Alg. 6.4	4.3e+0	1.1e+1	0	17.8	47.0
sensitivity	σ	2.3e-1	3.1e-1	0	1.0	1.0

TABLE 5.9. *Results for 100 samples (24) for $n = 100$, $k = 20$*

Frequently Pellet's bound is the best result; however, it fails for higher degrees due to the huge binomial coefficients to be used in the shifted polynomial Q. Neumaier's methods based on Gershgorin circles, which work with the original polynomial, show no failures and reasonable results in all cases.

In our next test set we generate a root cluster of different radius as follows:

$$P = \prod_{\nu=1}^{k} (x - \widetilde{z}(1 + \widetilde{r}_\nu \cdot e)) \cdot (x^{n-k} + \sum_{\nu=0}^{n-k-1} r_\nu x^\nu) \qquad \text{for } \widetilde{z} = 2, \tag{25}$$

where the $\widetilde{r}_\nu$ are again uniformly distributed random numbers in $[-1, 1]$. The coefficients of P calculated in floating point arithmetic are usually not exactly representable. This does not much harm; we may expect P to have a k-fold root cluster near $\widetilde{z}$ of radius $\max(e, \sigma)$, where σ denotes the sensitivity of $\widetilde{z}$ according to (14). In the following we display results for fixed $n = 20$, $k = 3$, and different values of e.

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	1.8e+0	1.8e+0	0	47198.9	65216.4
Montel	2	4.7e-1	4.8e-1	0	12665.6	17580.9
van Vleck	3	2.7e-4	4.6e-4	0	7.4	9.5
Neumaier-Gershgorin	4	3.5e-4	2.3e-3	0	9.2	69.8
Neumaier-refinement	5	6.1e-5	3.3e-4	0	1.6	10.3
Pellet	6	2.6e-5	4.4e-5	0	0.7	0.9
Neumaier-Rouché	7	7.3e-5	4.2e-4	0	1.9	13.4
factor	8	6.2e-5	1.8e-4	0	1.7	3.2
eigenvalues	9	2.9e-5	5.0e-5	0	0.8	1.0
hybrid with k	10	2.6e-5	4.4e-5	0	0.7	0.9
hybrid	Alg. 6.4	2.6e-5	4.4e-5	0	0.7	0.9
sensitivity	σ	3.7e-5	5.7e-5	0	1.0	1.0

TABLE 5.10. *Results for 100 samples (25) for* $n = 20$, $k = 3$ *and* $e = 10^{-10}$

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.4e+0	2.6e+0	0	70349.6	98600.1
Montel	2	5.9e-1	6.4e-1	0	16974.1	23758.4
van Vleck	3	2.7e-4	4.4e-4	0	7.8	10.3
Neumaier-Gershgorin	4	3.2e-4	6.5e-3	0	8.8	162.4
Neumaier-refinement	5	5.9e-5	6.9e-4	0	1.6	17.2
Pellet	6	2.6e-5	4.2e-5	0	0.7	1.0
Neumaier-Rouché	7	6.8e-5	1.1e-3	0	1.8	26.5
factor	8	5.9e-5	1.5e-4	0	1.8	2.7
eigenvalues	9	2.8e-5	4.6e-5	0	0.8	1.0
hybrid with k	10	2.6e-5	4.2e-5	0	0.7	1.0
hybrid	Alg. 6.4	2.6e-5	4.2e-5	0	0.7	1.0
sensitivity	σ	3.4e-5	5.4e-5	0	1.0	1.0

TABLE 5.11. *Results for 100 samples (25) for* $n = 20$, $k = 3$ *and* $e = 10^{-5}$

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	3.2e+0	3.4e+0	0	85927.6	128364.3
Montel	2	7.5e-1	8.2e-1	0	20521.0	30609.7
van Vleck	3	7.8e-4	1.4e-3	0	20.2	47.0
Neumaier-Gershgorin	4	3.2e-4	3.9e-3	0	8.4	93.4
Neumaier-refinement	5	7.8e-5	4.5e-4	0	2.1	10.6
Pellet	6	6.3e-5	1.1e-4	0	1.6	3.8
Neumaier-Rouché	7	8.6e-5	3.6e-4	0	2.3	8.6
factor	8	8.4e-5	1.7e-4	0	2.3	3.9
eigenvalues	9	6.3e-5	1.1e-4	0	1.7	3.8
hybrid with k	10	6.3e-5	1.1e-4	0	1.6	3.8
hybrid	Alg. 6.4	6.7e-5	1.2e-3	0	1.6	42.0
sensitivity	σ	3.6e-5	5.7e-5	0	1.0	1.0

TABLE 5.12. *Results for 100 samples (25) for* $n = 20$, $k = 3$ *and* $e = 10^{-4}$

We observe no failure of the methods and, for methods 3 to 10, results of the order of the theoretical sensitivity σ. The results in Table 5.12 are poorer than the others. This is due to the fact that the size of

the cluster already exceeds the sensitivity σ so that the roots are more k simple roots rather than a cluster. Especially the hybrid algorithm 6.4 detects different values of k in some instances.

Finally we generate two nearby k-fold roots as follows:

$$P = (x-\widetilde{z})^k(x-\widetilde{z}-e)^k(x^{n-2k}+\sum_{\nu=0}^{n-2k-1} r_\nu x^\nu) \qquad \text{for } \widetilde{z}=2 \tag{26}$$

with uniformly distributed random numbers r_ν in $[-1,1]$. Each polynomial P has an exact k-fold root at $\widetilde{z}$ and at $\widetilde{z}+e$, where the random numbers are generated with "few nonzero leading bits" to assure that this is also true when the coefficients of P are computed in floating point arithmetic. We give results for different values of e and fixed $n=20$, $k=3$.

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.0e+0	2.2e+0	0	6462.1	8766.6
Montel	2	5.2e-1	5.4e-1	0	1657.2	2199.1
van Vleck	3	2.2e-3	3.4e-3	0	6.7	7.9
Neumaier-Gershgorin	4	3.4e-3	1.4e-2	0	9.8	40.5
Neumaier-refinement	5	5.5e-4	1.7e-3	0	1.6	4.9
Pellet	6	2.1e-4	3.2e-4	0	0.6	0.8
Neumaier-Rouché	7	6.3e-4	2.1e-3	0	1.9	6.0
factor	8	2.1e-3	2.8e-3	0	6.3	6.9
eigenvalues	9	2.5e-4	3.6e-4	12	0.8	1.0
hybrid with k	10	2.1e-4	3.2e-4	0	0.6	0.8
hybrid	Alg. 6.4	2.1e-4	3.2e-4	0	0.6	0.8
sensitivity	σ	3.2e-4	4.8e-4	0	1.0	1.0

TABLE 5.13. *Results for 100 samples (26) for $n=20$, $k=3$ and $e=1/2$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.1e+0	2.3e+0	0	3646.9	4638.7
Montel	2	5.2e-1	5.4e-1	0	897.1	1131.2
van Vleck	3	4.0e-3	6.8e-3	0	6.7	7.7
Neumaier-Gershgorin	4	6.3e-3	9.0e-2	0	10.3	127.2
Neumaier-refinement	5	1.0e-3	1.2e-2	0	1.7	18.9
Pellet	6	3.8e-4	6.5e-4	0	0.6	0.7
Neumaier-Rouché	7	1.2e-3	1.4e-2	0	1.9	20.2
factor	8	5.2e-3	7.9e-3	37	10.0	12.7
eigenvalues	9	–	–	100	–	–
hybrid with k	10	3.8e-4	6.5e-4	0	0.6	0.7
hybrid	Alg. 6.4	3.8e-4	6.5e-4	0	0.6	0.7
sensitivity	σ	5.8e-4	9.0e-4	0	1.0	1.0

TABLE 5.14. *Results for 100 samples (26) for $n=20$, $k=3$ and $e=1/4$*

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.2e+0	2.5e+0	0	1867.4	2532.9
Montel	2	5.3e-1	5.7e-1	0	443.2	589.6
van Vleck	3	8.1e-3	1.5e-2	0	6.7	7.8
Neumaier-Gershgorin	4	1.4e-2	1.5e-1	0	11.5	100.7
Neumaier-refinement	5	2.3e-3	8.8e-2	0	1.8	60.1
Pellet	6	7.8e-4	1.4e-3	0	0.6	0.8
Neumaier-Rouché	7	2.6e-3	1.6e-2	0	2.0	10.8
factor	8	–	–	100	–	–
eigenvalues	9	–	–	100	–	–
hybrid with k	10	7.8e-4	1.4e-3	0	0.6	0.8
hybrid	Alg. 6.4	7.8e-4	1.4e-3	0	0.6	0.8
sensitivity	σ	1.2e-3	2.0e-3	0	1.0	1.0

TABLE 5.15. *Results for 100 samples (26) for* $n = 20$, $k = 3$ *and* $e = 1/8$

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.6e+0	2.9e+0	0	569.1	759.0
Montel	2	5.7e-1	6.4e-1	0	128.3	169.5
van Vleck	3	3.0e-2	5.4e-2	0	7.1	9.8
Neumaier-Gershgorin	4	6.9e-2	2.9e-1	0	14.4	43.5
Neumaier-refinement	5	2.7e-2	7.8e-2	0	6.1	15.3
Pellet	6	3.1e-3	5.7e-3	10	0.8	1.1
Neumaier-Rouché	7	1.0e-2	1.5e-2	46	2.2	3.6
factor	8	–	–	100	–	–
eigenvalues	9	–	–	100	–	–
hybrid with k	10	3.2e-3	5.0e-2	0	0.8	7.1
hybrid	Alg. 6.4	3.2e-3	5.0e-2	0	0.8	7.1
sensitivity	σ	4.5e-3	7.4e-3	0	1.0	1.0

TABLE 5.16. *Results for 100 samples (26) for* $n = 20$, $k = 3$ *and* $e = 1/32$

		radius			ratio to sensitivity	
method		median	max	failed	median	max
Ostrowski-R.	1	2.8e+0	3.0e+0	0	373.8	552.1
Montel	2	6.4e-1	6.7e-1	0	83.6	123.5
van Vleck	3	1.2e-1	1.8e-1	0	16.1	24.5
Neumaier-Gershgorin	4	6.7e-2	8.7e+0	0	8.8	520.9
Neumaier-refinement	5	2.3e-2	8.7e+0	0	2.9	520.9
Pellet	6	–	–	100	–	–
Neumaier-Rouché	7	–	–	100	–	–
factor	8	–	–	100	–	–
eigenvalues	9	–	–	100	–	–
hybrid with k	10	2.3e-2	1.2e-1	0	2.9	9.3
hybrid	Alg. 6.4	1.1e-2	1.6e-2	0	1.5	2.0
sensitivity	σ	7.5e-3	1.7e-2	0	1.0	1.0

TABLE 5.17. *Results for 100 samples (26) for* $n = 20$, $k = 3$ *and* $e = 1/128$

Especially in the last tables we see the advantages of methods 1, 2, 3, 4 and 5: they always deliver a disc containing at least k roots of P. Methods 4 and 5 give the additional advantage of an exact root counting.

Next we come to the interpretation of the results. First we observe that Method 1 (based on a modification of Ostrowski's theorem) is in all tests worse than Method 2 (based on Montel's theorem) by a factor of 3 to 4, and in turn Method 2 is sometimes by several orders of magnitude worse than Method 3 (based on van Vleck's theorem). The main reason for the poor behavior of Methods 1 and 2 is that the highest term in the polynomials in Theorems 2.4 and 2.5 is z^n implying a root bound proportional to the n-th rather than the k-th root of certain quantities.

For smaller degrees Method 3 seems to be superior Method 4 based on Neumaier's version of Gershgorin circles, whereas for wide clusters and especially for larger degrees Method 4 gives better results. Note that Method 4 and the correction in Method 5 determines the number of roots near $\widetilde{z}$ by itself. This can be useful as "backup" in our hybrid method. Therefore we decided to use Method 3 as a backup and Methods 4 and 5 as a second backup in our hybrid Method 10 in case other methods fail. Care has to be taken, however, because Method 3 shows some huge bounds due to small values of q_k.

The remaining methods are sufficient criterions or other kind of methods with the potential to fail. The first Methods 6 (based on Pellet's theorem) shows generally very good results. Therefore we choose to use Pellet's method as the first one in our hybrid method.

If Method 6 does not fail, the result is always better than those of Methods 3 and 4, rarely the refined bounds by Method 5 are a little better than those by Method 6. In fact Method 6 delivers almost always the best results except for simple roots and for large values of n or k. In that case Method 7 (based on Neumaier's application of Rouché's theorem) is usually better than Method 5. For simple roots we will use a still better method in our hybrid method to be described in the next section. For larger values of n we use Method 7 as a third backup.

The self-validating Method 8 (based on error bounds for a factor of degree k) is worse than Method 6 by up to one order of magnitude for smaller degree, but superior for larger values of n. Method 9 (based on inclusion of eigenvalues of a companion matrix) delivers results of about the same quality as Method 6 for smaller degree and is again better for larger values of n. However, this is achieved with significantly increased computational effort.

We want to stress that our judgement is based on a limited number of test cases. Although we tried to cover the major important situations which may occur in the realm of root clusters of polynomials, it may happen that for other kinds of examples the situation changes significantly. This is a reason why we present a variety of methods, although some of them seem to show a poorer behavior than others.

6. A new hybrid method. We want to gather the experience with the previous methods to devise a new hybrid method for computing bounds for multiple roots of polynomials. In contrast to the previous methods (except Methods 4 and 5) the number of roots of the cluster need not to be specified in advance but will be detected by the algorithm. The aim is to design a hybrid method which always determines an enclosing disc, ideally of the best of sizes delivered by any of the other methods. The only information we use as input is the polynomial P and some approximation $\widetilde{z}$ near which a root cluster is looked for.

In case of a root cluster the number k is of course not uniquely determined, it may depend on what is considered to belong to the cluster and what not, and of course also on when a root may be considered as simple rather than a member of a cluster. Therefore we have to rely on some heuristic in the determination of k trying to imitate the judgement of a human observer. This turned out to be not that easy. We put some effort into that question because an inappropriate value for k may alter the computed bounds significantly.

A first idea may be to look at the coefficients $q_0, q_1, \ldots$ of Q as in (1). For a three-fold root, for example, typical values would be $|q_0| = 10^{-9}$, $|q_1| = 10^{-6}$, $|q_2| = 10^{-3}$, $|q_\nu| \sim \mathcal{O}(1)$ for $\nu \geq 3$. So one might look for the first index k for which $|q_{k+1}/q_k|$ is small. However, we did not manage to translate this idea into a numerically reliable criterion.

Another idea would be to calculate approximations z_ν for the roots of P, sorted by distance from $\widetilde{z}$ and look for a gap. However, this did not work very well because it depends on the quality of $\widetilde{z}$. A better approach is to use the fact that the nonnegative root of W as in (17) is a good approximation to a potential radius of an enclosing disc by Method 6. So one may take the value of k with smallest such root. This gives very good results but, especially for two root clusters, these are still not satisfactory.

Our determination of k works as follows. We use the sensitivity σ as in (14), which turned out to be a good measure for the clustering of roots near $\widetilde{z}$. Given P, $\widetilde{z}$ and approximations z_ν to the roots of P we proceed as follows.

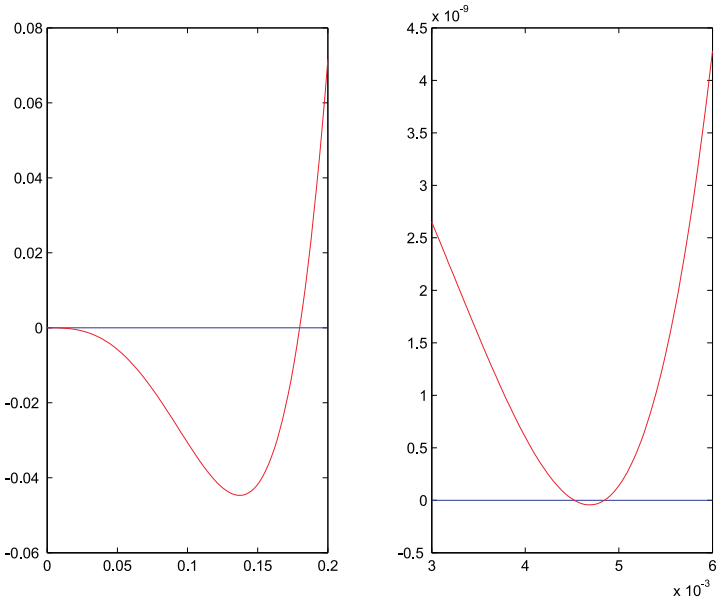
$k = 1$; $\varepsilon = 2^{-52}$;
for $m = 1 : n$
  $\sigma = (\varepsilon \cdot |P|(|\widetilde{z}|)/|P^{(m)}(\widetilde{z})|/m! \,)^{1/m}$;
  if $|z_\nu - \widetilde{z}| < 2\sigma$ for m values of ν, set $k := m$ and return
end

ALGORITHM 6.1. *Determination of the number of roots near* $\widetilde{z}$

In our experience this simple algorithm shows good results. In almost all cases the heuristic produces the same k as a human observer. The computational effort is bounded by approximately $3(n-k)k$ operations, where k denotes the determined number of roots near $\widetilde{z}$.

Based on the value of k we compute a disc $D(c, r)$ containing exactly or at least k roots of P. If $k = 1$ we assume a simple root. For this case very good and well known self-validating procedures are available. We choose the algorithm `verifynlss`, a general nonlinear system solver available in INTLAB [30], which is based on the application of Krawczyk's operator (21) with various improvements described in [29]. This algorithm is applied to the nonlinear equation $P(z) = 0$. We rarely encountered a case where this procedure failed for the value $k = 1$ computed by Algorithm 6.1. If it fails, we return a disc D with radius r which is the minimum of (7) and (9).

For $k \geq 2$ we calculate an enclosing disc D by Method 6. For this we improve the determination of r for exceptional situations. The normal case is shown in Graph 3.4, for which Method 6 works good. For increasing radius of the root cluster or for decreasing separation from the other roots the nonnegative root of W is still a good lower bound for the smaller root of V; however, we then have the problem to find an upper bound for the smaller root of V. Such a situation may look as follows.



GRAPH 6.2. *Normal and exceptional behavior of* V

For the "normal" behavior on the left (similar to Graph 3.4) we have a very small and a large positive root of V; in exceptional situations as on the right we may find two close positive roots of P. It is easy to see by Descarte's rule of sign that if V has too nonnegative roots, so does $V', V'', \ldots, V^{(k-1)}$. Therefore, if the guess for r in Method 6 fails, we approximate the root of V' near r by some Newton steps. Then, after one secant step, we usually have a starting point for which a Newton iteration for V converges. If not, the following backup algorithm is applied.

Normally Method 3 always yields an enclosing disc. The method fails if $q_k = 0$, that is the guess for k was unsatisfactory. The method may deliver unsatisfactory bounds for large values of n because of inaccuracies in the computation of the coefficients of Q. In either case we use Methods 4 and 7 as backup because they rely on the Durand-Kerner corrections α_ν. This backup may only fail if approximations z_ν are so close that $z_\nu - z_\mu$ becomes numerically zero. We summarize this backup approach by the following algorithm. We mention that this second backup was never necessary in our computational tests.

1) Compute $D = D(c, r)$ by Method 3 and σ by (14). If successful and $r < 2\sigma$, return D.
2) Compute D_1 by Methods 4 and 5, and D_2 by Method 7.
   If both methods fail, return "no result".
   Otherwise return disc with smallest radius.

ALGORITHM 6.3. *Backup solution for otherwise non-solvable situations*

Summarizing, this defines the following hybrid algorithm, which tries to combine the advantages of five of the previous methods: It mainly uses Pellet's test with specific detection of the radius, it uses a self-validating method for simple roots and, as a backup, the van Vleck, Neumaier's Gershgorin with correction and Neumaier's Rouché bound.

1) Compute k by Algorithm 6.1.
2) If $k = 1$, call `verifynlss`, else goto 3).
   If success, return, else compute enclosing disc by the minimum of (7) and (9) and return.
3) Compute c, Q according to Algorithm 2.1 and (1), respectively.
   if $q_k = 0$, goto 8).
   if $q_0 = \ldots = q_{k-1}$, return $D(c, 0)$. [exact k-fold root]
4) Compute W according to (17) and compute r by some Newton iterations starting at the Fujiwara root bound for W.
5) Perform one damped Newton iteration on V as in (16) starting at r and resulting in R.
   [Usually, R is an upper bound for the final radius.]
   if $R \le 0$, goto 8) [no bound by Pellet's criterion].
6) If $V(R) > 0$, perform some Newton iterations on V' starting at R resulting in r. Perform one secant step on V in $[R, r]$ resulting in R.
   If $V(R) > 0$, goto 8) [no bound by Pellet's criterion].
7) [the disc $D(c, R)$ contains exactly k roots of P]
   Improve R by some Newton iterations on V until Pellet's criterion is no longer satisfied.
   Return $D(c, R)$.
8) [backup]
   Compute D by Algorithm 6.3 and return.

ALGORITHM 6.4. *Hybrid algorithm for computation of k and an enclosing disc D*

We now return to the numerical results of the previous section. The third-last row "Method 10" in the tables depicts the results of Algorithm 6.4 for prespecified k (starting in step 2). We notice that in this case the hybrid algorithms almost always gives the best of all results of Methods 1 to 9. This is also the case for Table 5.17. The displayed numbers of Method 10 are averaged over *all* 100 samples whereas those of the other methods only over the nonfailing samples. Especially for simple roots the results of the hybrid algorithm sometimes beats the other ones by almost an order of magnitude.

For the hybrid algorithm we tried to find a reasonable compromise between performance and quality of the bounds. Therefore, the results of the hybrid algorithm are not always the best ones compared to the other methods. For example, the Methods 4/5 and 7 are only applied in the backup Algorithm 6.3 if van Vleck's method fails or the computed radius is significantly above the sensitivity. Table 5.9 with $n = 100$ and $k = 20$ displays the limits of the approach. Here the sensitivity of the root cluster relative to the separation to the other roots is just too big, so that only the backup is used and all roots are enclosed.

The results of the hybrid Algorithm 6.4 with determination of k in step 1) yields identical or almost identically results compared to the version with specified value of k. Exceptions are the tests in Tables 5.12 and 5.17. In both cases clustered roots are separated by about the size of the sensitivity. In the first case of Table 5.12 the roots are scattered near $\widetilde{z}$ and a different value for k yields sometimes poorer results. For the two k-fold roots in Table 5.17 the algorithm switches to $2k$ roots near $\widetilde{z}$ if they are close enough and better results are delivered. In both cases Algorithm 6.4 determined a different value for k: 60 times in the examples of Table 5.12 and always for the samples of Table 5.17. Otherwise this happened only for $n = 100$ and $k = 20$ in Table 5.9, where indeed a 20-fold root cluster is hardly recognizable. We stress again that in all cases i) the hybrid algorithm does compute an enclosing disc with ii) a radius of the order of the sensitivity σ.

Finally, we measured the computational effort of the hybrid algorithm. The following table specifies the median and maximum number of polynomial evaluations of Algorithm 6.4 (including determination of k) for the test sets according to the tables of the previous section.

				polynomial evaluations	
Table	n	k	e	median	maximum
5.1 ... 3	20/40/100	3	0	20/30/168	20/30/170
5.4 ... 6	40	1/2/5	0	6/28/35	45/28/35
5.7 ... 9	100	1/5/20	0	6/168/207	105/170/268
5.10 ... 12	20	3	$10^{-10}/10^{-5}/10^{-4}$	20/20/28	20/23/53
5.13 ... 17	20	3	$2^{-1}/2^{-2}/2^{-3}/2^{-5}/2^{-7}$	20/20/20/26/28	20/20/23/52/31

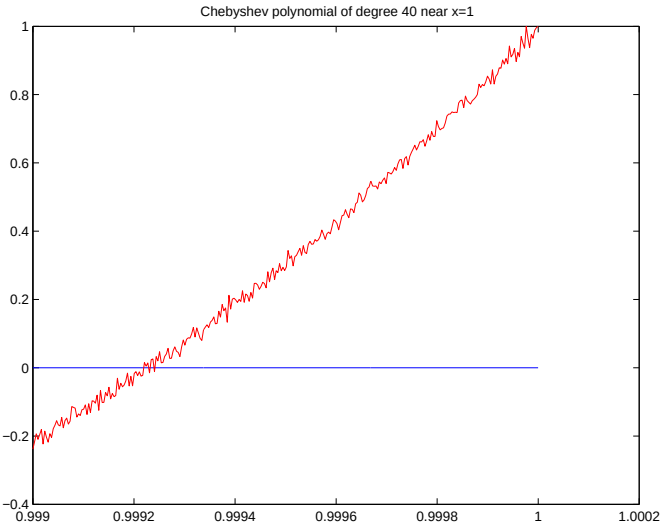
TABLE 6.5. *Median and maximum number of polynomial evaluations of Algorithm 6.4*

For moderate sensitivities there seems a moderate dependence on n and k. For larger values of n and k and for increasing sensitivity we observe certain jumps and superlinear behavior due to the various backup strategies in Algorithm 6.3. This is the prize we paid for fairly stable numerical results. Anomalies due to particular random test cases like in the last row for $e = 2^{-5}$ and $e = 2^{-7}$ may occur, but turned out to be exceptional when repeating the tests. The numbers suggest a total computing time of the hybrid algorithm of the order of n^2.

The hybrid Algorithm 6.4 is implemented in Matlab and included in Version 4 of the Matlab interval toolbox INTLAB [30]. It can be downloaded from our homepage (cf. [30]). Finally we mention that the restricted precision of floating point arithmetic imposes certain limits. Consider, for example, the Chebyshev polynomial T_{40} of degree 40. Calculating an inclusion of the root near $x = 1$ by our hybrid Algorithm 6.4 (without specifying multiplicity) yields the inclusion

$$X = [0.9953, 1.0031] \tag{27}$$

with the proof that X contains at least on root of T_{40}. One may argue that this is a poor inclusion. In fact, the verification algorithm in step 2) of Algorithm 6.4 failed and the bound is the one obtained by (9). The maximum absolute value of the coefficients of T_{40} is $2.1 \cdot 10^{14}$ which causes quite some cancellation when evaluating the polynomial in double precision ($\varepsilon = 2.2 \cdot 10^{-16}$) near $x = 1$, and the sensitivity $2.2 \cdot 10^{-4}$ of the largest positive root 0.9992 of T_{40} corresponds to the quality of the inclusion (27). Indeed, the graph near $x = 1$ displays the effects of rounding and cancellation.



GRAPH 6.6. *Floating point evaluation of T_{40} near $x = 1$.*

REFERENCES

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York, 1983.

[2] P. Batra. *Abschätzungen und Iterationsverfahren für Polynom-Nullstellen.* Dissertation, Technische Informatik III, TU Hamburg-Harburg, 1999.

[3] H.E. Bell. Gershgorin's Theorem and the zeros of polynomials. *Amer. Math. Monthly*, 72:292–295, 1965.

[4] R. Bhatia, L. Elsner, and G. Krause. Bounds for the variation of the roots of a polynomial and the eigenvalues of a matrix. *Linear Algebra Appl.*, 142:195–209, 1990.

[5] W. Börsch-Supan. Residuenabschätzungen für Polynom-Nullstellen mittels Lagrange-Interpolation. *Numer. Math.*, 14:287–296, 1970.

[6] D. Braess and K.P. Hadeler. Simultaneous inclusion of the zeros of a polynomial. *Numer. Math.*, 21:161–165, 1973.

[7] C. Carstensen. Inclusion of the roots of a polynomial based on Gershgorin's theorem. *Numer. Math.*, 59:349–360, 1991.

[8] C. Carstensen and E. Stein. Über die Falksche ECP-Transformation und Verallgemeinerungen. *Z. Angew. Math. Mech.*, 69:375–391, 1989.

[9] I.E. Durand. Solutions Numérique des Équations Algébriques, Tome I: Équations du Type $F(x) = 0$. *Racines d'une Polynôme Masson*, pages 279–281, 1960.

[10] L. Elsner. A remark on simultaneous inclusions of the zeros of a polynomial by Gershgorin's theorem. *Numer. Math.*, 21:425–427, 1973.

[11] A. Frommer. Proving Conjectures by Use of Interval Arithmetic. In U. Kulisch et al., editor, *Perspectives on enxlosure methods. SCAN 2000, GAMM-IMACS international symposium on scientific computing, computer arithmetic and validated numerics, Univ. Karlsruhe, Germany, September 19-22, 2000*, Wien, 2001. Springer.

[12] I. Gargantini and P. Henrici. Circular Arithmetic and the Determination of Polynomial Zeros. *Numer. Math.*, 18:305–320, 1972.

[13] T.C. Hales. The Kepler conjecture. Manuscript, 1998. `http://www.math.lsa.umich.edu/~hales/countdown/`.

[14] J. Hass, M. Hutchings, and R. Schlafly. The Double Bubble Conjecture. *Elec. Res. Announcement of the Am. Math. Soc.*, 1, No. 3:98–102, 1995.

[15] P. Henrici. *Applied and computational complex analysis, Vol. 1.* Power Series - Integration - Conformal Mapping - Location of Zeros. Wiley, New York, 1974.

[16] *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*, 1985.

[17] I.O. Kerner. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math*, 8:290–294, 1966.

[18] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4:187–201, 1969.

[19] M. Marden. Geometry of Polynomials. *AMS Mathematical Surveys 3*, 1966.

[20] MATLAB User's Guide, Version 6. The MathWorks Inc., 2000.

[21] J.M. McNamee. A bibliography on roots of polynomials. *J Comput Appl. Math.*, 47:391–394, 1993.

[22] R.E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1966.

[23] A. Neumaier. Enclosing clusters of zeros of polynomials. to appear.

[24] A. Neumaier. An Existence Test for Root Clusters and Multiple Roots. *ZAMM*, 68(6):256–257, 1988.

[25] A. Neumaier. *Interval Methods for Systems of Equations.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[26] A. Ostrowski. *Solution of Equations in Euclidean and Banach Spaces.* Academic, New York, 3rd edition, 1973.

[27] D. Phillips. Improving spectral-variation bounds with Chebyshev polynomials. *Linear Algebra Appl.*, 133:165–173, 1990.

[28] L.B. Rall. *Automatic differentiation: techniques and applications*, volume 120 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg-New York, 1981.

[29] S.M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics*, pages 63–136, Elsevier, Amsterdam, 1994.

[30] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tu-harburg.de/english/index.html`.

[31] S.M. Rump. Self-validating methods. *Linear Algebra and its Applications (LAA)*, 324:3–13, 2001.

[32] A. Schönhage. Quasi-gcd computations. *J. Complexity*, 1:118–137, 1985.

[33] B.T. Smith. Error bounds for zeros of a polynomial based upon Gerschgorin's theorems. *J. ACM*, 17:661–674, 1970.

[34] T. Sunaga. Theory of an Interval Algebra and its Application to Numerical Analysis. *RAAG Memoirs*, 2:29–46, 1958.

[35] W. Tucker. The Lorenz attractor exists. *C. R. Acad. Sci., Paris, Sér. I, Math.*, 328(12):1197–1202, 1999.

[36] A. van der Sluis. Upper bounds for roots of polynomials . *Numer. Mathematik*, 15:250–62, 1970.

[37] K. Weierstraß. *Neuer Beweis des Satzes, daß jede ganze rationale Function einer Veränderlichen dargestellt werden kann als ein Product aus linearen Functionen derselben Veränderlichen*, volume 3 of *Ges. Math. Werke*, pages 251–269. Meyer und Muller, Berlin, 1903. (Reprint by Johnson Reprint Corp., New York, 1967).

[38] W. Werner. Über Abschätzungen von Polynomnullstellen mittels des Gerschgorinschen Kreissatzes. *Z. Angew. Math. Mech.*, 63:T390–391, 1983.

[39] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.